NewCleanProject

Dieudonne Ouedraogo

5/6/2017

### Namimg columns;The dataset columns could be named easily in R with the code below

### Mice is an imputation package ,that uses heuristic algorithm to imput missing values in a dataset without inserting much bias.

### Caret is a machine learning package wildy use in R for machine learning techniques

library(mice)  
library(readr)  
library(caret)  
BreastCancer<- read\_csv("~/Downloads/breast-cancer-wisconsin.data.txt",col\_names = FALSE)  
colnames(BreastCancer) <- c("SampleCodeNumber",   
 "ClumpThickness",   
 "UniformityOfCellSize",   
 "UniformityOfCellShape",   
 "MarginalAdhesion",   
 "SingleEpithelialCellSize",   
 "BareNuclei",   
 "BlandChromatin",   
 "NormalNucleoli",   
 "Mitosis",   
 "Classes")  
#write.csv(BreastCancer, 'BreastCancerWisconsin2.csv', row.names=FALSE, quote=FALSE)

### Separating Classes based on the value of Benign and MalignantIn the dataset we only have 2 and 4 as number indicating Benign and Malignant:We try to change that in the code below ,anything else other than (2,4) we replace by NA

BreastCancer$Classes <- ifelse(BreastCancer$Classes == "2", "Benign",  
 ifelse(BreastCancer$Classes == "4", "Malignant", NA))  
#Data cleaning   
BreastCancer[BreastCancer == "?"] <- NA  
  
#HERE I SAVE A CLEAN VERSION WITH HEADERS  
#write.csv(BreastCancer, 'BreastCancerWisconsin3.csv', row.names=FALSE, quote=FALSE)

### Count of total NA

length(which(is.na(BreastCancer)))  
nrow(BreastCancer[is.na(BreastCancer), ])  
nrow(BreastCancer)

### CORRELATION BETWEEN VARIABLE;WE USE MICE FOR IMPUTATION :FILLING MISSING VALUES WITH APPROPRIATE VALUES ;WE COULD HAD DROP THE MISSING VALUES ,BUT SINCE THE DATASET IS NOT THAT LARGE IT CAN AFFECT OUR RESULTS. WE INPUTE ,BUT INPUTING RAISE INCERTAINTY;WE USE MICE (WHICH REDUCE INCERTAINTY )

library(corrplot)  
BreastCancer[,2:10] <- apply(BreastCancer[, 2:10], 2, function(x) as.numeric(as.character(x)))  
dataset\_impute <- mice(BreastCancer[, 2:10], print = FALSE)  
BreastCancer <- cbind(BreastCancer[, 11, drop = FALSE], mice::complete(dataset\_impute, 1))  
  
BreastCancer$Classes <- as.factor(BreastCancer$Classes)  
#write.csv(BreastCancer, 'BreastCancerWisconsinComplete.csv', row.names=FALSE, quote=FALSE)  
pander::pander(cor(BreastCancer[, 2:10]))

Table continues below

|  |  |  |
| --- | --- | --- |
|  | ClumpThickness | UniformityOfCellSize |
| **ClumpThickness** | 1 | 0.6449 |
| **UniformityOfCellSize** | 0.6449 | 1 |
| **UniformityOfCellShape** | 0.6546 | 0.9069 |
| **MarginalAdhesion** | 0.4864 | 0.7056 |
| **SingleEpithelialCellSize** | 0.5218 | 0.7518 |
| **BareNuclei** | 0.5943 | 0.6952 |
| **BlandChromatin** | 0.5584 | 0.7557 |
| **NormalNucleoli** | 0.5358 | 0.7229 |
| **Mitosis** | 0.35 | 0.4587 |

Table continues below

|  |  |  |
| --- | --- | --- |
|  | UniformityOfCellShape | MarginalAdhesion |
| **ClumpThickness** | 0.6546 | 0.4864 |
| **UniformityOfCellSize** | 0.9069 | 0.7056 |
| **UniformityOfCellShape** | 1 | 0.6831 |
| **MarginalAdhesion** | 0.6831 | 1 |
| **SingleEpithelialCellSize** | 0.7197 | 0.5996 |
| **BareNuclei** | 0.7159 | 0.671 |
| **BlandChromatin** | 0.7359 | 0.6667 |
| **NormalNucleoli** | 0.7194 | 0.6034 |
| **Mitosis** | 0.4389 | 0.4176 |

Table continues below

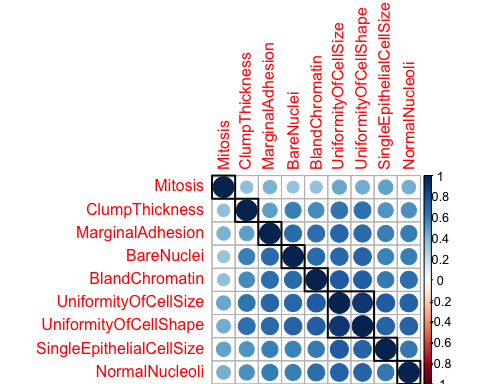
|  |  |  |
| --- | --- | --- |
|  | SingleEpithelialCellSize | BareNuclei |
| **ClumpThickness** | 0.5218 | 0.5943 |
| **UniformityOfCellSize** | 0.7518 | 0.6952 |
| **UniformityOfCellShape** | 0.7197 | 0.7159 |
| **MarginalAdhesion** | 0.5996 | 0.671 |
| **SingleEpithelialCellSize** | 1 | 0.5875 |
| **BareNuclei** | 0.5875 | 1 |
| **BlandChromatin** | 0.6161 | 0.6826 |
| **NormalNucleoli** | 0.6289 | 0.5926 |
| **Mitosis** | 0.4791 | 0.3359 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | BlandChromatin | NormalNucleoli | Mitosis |
| **ClumpThickness** | 0.5584 | 0.5358 | 0.35 |
| **UniformityOfCellSize** | 0.7557 | 0.7229 | 0.4587 |
| **UniformityOfCellShape** | 0.7359 | 0.7194 | 0.4389 |
| **MarginalAdhesion** | 0.6667 | 0.6034 | 0.4176 |
| **SingleEpithelialCellSize** | 0.6161 | 0.6289 | 0.4791 |
| **BareNuclei** | 0.6826 | 0.5926 | 0.3359 |
| **BlandChromatin** | 1 | 0.6659 | 0.3442 |
| **NormalNucleoli** | 0.6659 | 1 | 0.4283 |
| **Mitosis** | 0.3442 | 0.4283 | 1 |

correlations =cor(BreastCancer[, 2:10])

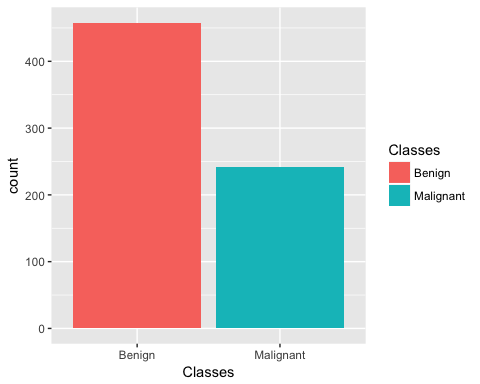
### We use package corrplot to visually see the correlation between variables ,darker means more correlated

library(corrplot)  
corrplot(correlations, order = "hclust", tl.cex=1, addrect = 8)



### We can group the cases and we can know the prior probabilities then NUMBER OF BENIGN AND MALIGNANT IN THE DATASET

library(ggplot2)  
ggplot(BreastCancer, aes(x = Classes, fill = Classes)) +  
 geom\_bar()



### We can decide to remove highly variable and use this as dimension reduction technique We use the function findCorrelation from the package Caret remove predictors based on whose correlation is above 0.85. This function uses a heuristic algorithm to determine which variable should be removed

highCorr <- findCorrelation(correlations, cutoff = .85)

### We find only 1 element ,obviously we could had seen that from the table of correlation;Uniformity of cell Size and uniformity of cell shape are higly correlated ,we can cut one off

highCorr

## [1] 2

length(highCorr)#Only 1 element has high correlation

## [1] 1

### Let's filter our dataset ,by removing one column,the third columns from breast cancer dataset

df=BreastCancer[, 2:10]  
filteredBcData <- BreastCancer[, -3]  
head(filteredBcData)

## Classes ClumpThickness UniformityOfCellShape MarginalAdhesion  
## 1 Benign 5 1 1  
## 2 Benign 5 4 5  
## 3 Benign 3 1 1  
## 4 Benign 6 8 1  
## 5 Benign 4 1 3  
## 6 Malignant 8 10 8  
## SingleEpithelialCellSize BareNuclei BlandChromatin NormalNucleoli  
## 1 2 1 3 1  
## 2 7 10 3 2  
## 3 2 2 3 1  
## 4 3 4 3 7  
## 5 2 1 3 1  
## 6 7 10 9 7  
## Mitosis  
## 1 1  
## 2 1  
## 3 1  
## 4 1  
## 5 1  
## 6 1

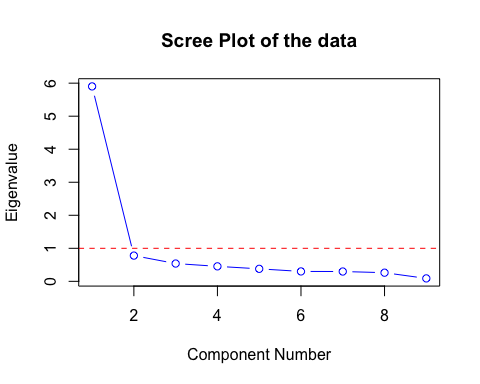
df2=filteredBcData[,-1]

### We can argue that since the variables have same scale ,it's not neccessary to mean corrected the data or scaled ;regarless we try to see below the effect of doing so define the transformation or pre-processing

bc.trans <- preProcess(filteredBcData, method = c("BoxCox", "center", "scale"))  
bc.transformed <- predict(bc.trans, filteredBcData)  
#pander::pander(head(bc.transformed))

### Principal Components analysis

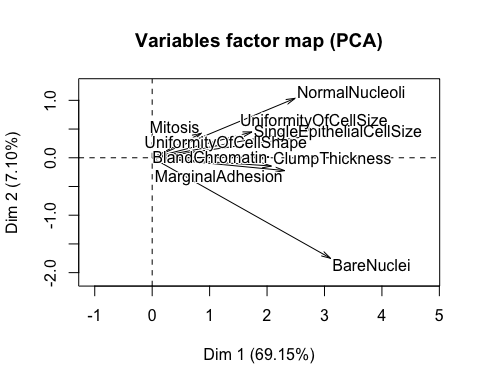
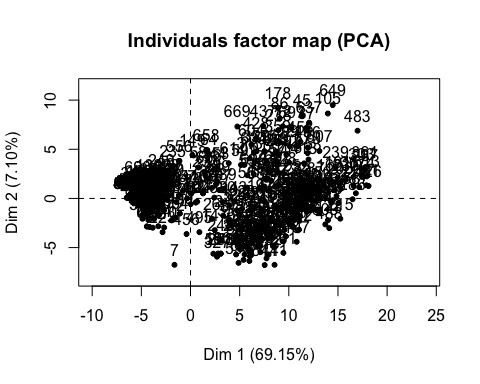
Scree.Plot <- function(R,main="Scree Plot",sub=NULL){  
 roots <- eigen(R)$values  
 x <- 1:dim(R)[1]  
 plot(x,roots,type="b",col='blue',ylab="Eigenvalue",  
 xlab="Component Number",main=main,sub=sub)   
 abline(h=1,lty=2,col="red")  
   
}  
R <- cor(df)  
R2=cor(df2)  
Scree.Plot(R,main ="Scree Plot of the data")



Scree.Plot(R2,main ="Scree Plot of the data")



library(FactoMineR)  
X=df  
X2=df2  
P=PCA(X,ncp=ncol(X),scale.unit=F,graph=T)



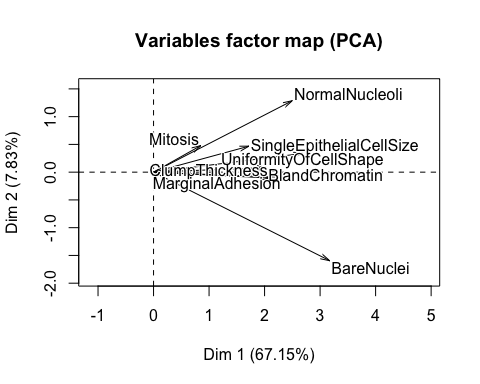
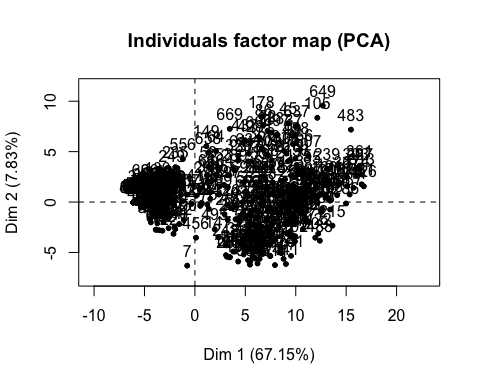
summary(P)

##   
## Call:  
## PCA(X = X, scale.unit = F, ncp = ncol(X), graph = T)   
##   
##   
## Eigenvalues  
## Dim.1 Dim.2 Dim.3 Dim.4 Dim.5 Dim.6  
## Variance 48.754 5.005 4.282 3.153 2.733 2.406  
## % of var. 69.152 7.099 6.073 4.472 3.876 3.412  
## Cumulative % of var. 69.152 76.251 82.325 86.797 90.673 94.085  
## Dim.7 Dim.8 Dim.9  
## Variance 1.777 1.591 0.802  
## % of var. 2.520 2.257 1.138  
## Cumulative % of var. 96.606 98.862 100.000  
##   
## Individuals (the 10 first)  
## Dist Dim.1 ctr cos2 Dim.2 ctr  
## 1 | 4.996 | -4.444 0.058 0.791 | 0.033 0.000  
## 2 | 7.996 | 4.865 0.069 0.370 | -4.735 0.641  
## 3 | 4.750 | -4.594 0.062 0.935 | -0.620 0.011  
## 4 | 8.384 | 5.137 0.077 0.375 | 3.413 0.333  
## 5 | 4.644 | -4.080 0.049 0.772 | -0.103 0.000  
## 6 | 15.413 | 15.045 0.664 0.953 | -0.468 0.006  
## 7 | 8.466 | -1.630 0.008 0.037 | -6.750 1.302  
## 8 | 5.201 | -4.944 0.072 0.904 | 0.395 0.004  
## 9 | 6.892 | -5.428 0.086 0.620 | 0.977 0.027  
## 10 | 4.838 | -4.629 0.063 0.916 | 0.325 0.003  
## cos2 Dim.3 ctr cos2   
## 1 0.000 | 1.736 0.101 0.121 |  
## 2 0.351 | -1.023 0.035 0.016 |  
## 3 0.017 | -0.051 0.000 0.000 |  
## 4 0.166 | 2.168 0.157 0.067 |  
## 5 0.000 | -0.071 0.000 0.000 |  
## 6 0.001 | -0.634 0.013 0.002 |  
## 7 0.636 | -2.449 0.200 0.084 |  
## 8 0.006 | -0.737 0.018 0.020 |  
## 9 0.020 | -0.721 0.017 0.011 |  
## 10 0.005 | 1.022 0.035 0.045 |  
##   
## Variables  
## Dim.1 ctr cos2 Dim.2 ctr cos2   
## ClumpThickness | 2.076 8.836 0.544 | -0.144 0.417 0.003 |  
## UniformityOfCellSize | 2.815 16.256 0.852 | 0.523 5.459 0.029 |  
## UniformityOfCellShape | 2.731 15.298 0.846 | 0.376 2.832 0.016 |  
## MarginalAdhesion | 2.307 10.913 0.653 | -0.224 1.000 0.006 |  
## SingleEpithelialCellSize | 1.739 6.200 0.617 | 0.450 4.038 0.041 |  
## BareNuclei | 3.106 19.782 0.728 | -1.750 61.201 0.231 |  
## BlandChromatin | 2.035 8.492 0.697 | 0.013 0.003 0.000 |  
## NormalNucleoli | 2.490 12.719 0.666 | 1.035 21.422 0.115 |  
## Mitosis | 0.856 1.504 0.250 | 0.426 3.628 0.062 |  
## Dim.3 ctr cos2   
## ClumpThickness 1.759 72.234 0.391 |  
## UniformityOfCellSize 0.057 0.076 0.000 |  
## UniformityOfCellShape 0.158 0.586 0.003 |  
## MarginalAdhesion -0.990 22.902 0.120 |  
## SingleEpithelialCellSize -0.096 0.215 0.002 |  
## BareNuclei -0.180 0.761 0.002 |  
## BlandChromatin -0.225 1.178 0.008 |  
## NormalNucleoli -0.289 1.952 0.009 |  
## Mitosis -0.064 0.096 0.001 |

dimdesc(P, axes=c(1,2,3))#PRINT THE FIRST 3 COMPONENTS

## $Dim.1  
## $Dim.1$quanti  
## correlation p.value  
## UniformityOfCellSize 0.9232495 8.952913e-292  
## UniformityOfCellShape 0.9195869 5.257034e-285  
## BareNuclei 0.8533332 2.496263e-199  
## BlandChromatin 0.8350859 4.537223e-183  
## NormalNucleoli 0.8160532 4.165803e-168  
## MarginalAdhesion 0.8083870 1.456650e-162  
## SingleEpithelialCellSize 0.7857455 1.488441e-147  
## ClumpThickness 0.7376715 5.106833e-121  
## Mitosis 0.4996251 2.061220e-45  
##   
##   
## $Dim.2  
## $Dim.2$quanti  
## correlation p.value  
## NormalNucleoli 0.33934334 2.680932e-20  
## Mitosis 0.24865878 2.603305e-11  
## SingleEpithelialCellSize 0.20318240 5.997757e-08  
## UniformityOfCellSize 0.17142055 5.169526e-06  
## UniformityOfCellShape 0.12677013 7.817167e-04  
## MarginalAdhesion -0.07839555 3.825149e-02  
## BareNuclei -0.48092119 9.665890e-42  
##   
##   
## $Dim.3  
## $Dim.3$quanti  
## correlation p.value  
## ClumpThickness 0.62503064 5.082569e-77  
## BlandChromatin -0.09218520 1.476552e-02  
## NormalNucleoli -0.09473514 1.221677e-02  
## MarginalAdhesion -0.34705066 3.244524e-21

P2=PCA(X2,ncp=ncol(X),scale.unit=F,graph=T)



summary(P2)

##   
## Call:  
## PCA(X = X2, scale.unit = F, ncp = ncol(X), graph = T)   
##   
##   
## Eigenvalues  
## Dim.1 Dim.2 Dim.3 Dim.4 Dim.5 Dim.6  
## Variance 41.102 4.790 4.279 3.048 2.449 2.273  
## % of var. 67.154 7.827 6.991 4.980 4.001 3.713  
## Cumulative % of var. 67.154 74.981 81.972 86.952 90.953 94.666  
## Dim.7 Dim.8  
## Variance 1.713 1.551  
## % of var. 2.799 2.535  
## Cumulative % of var. 97.465 100.000  
##   
## Individuals (the 10 first)  
## Dist Dim.1 ctr cos2 Dim.2 ctr  
## 1 | 4.517 | -3.914 0.053 0.751 | 0.055 0.000  
## 2 | 7.949 | 4.974 0.086 0.392 | -4.559 0.621  
## 3 | 4.243 | -4.072 0.058 0.921 | -0.577 0.010  
## 4 | 6.828 | 3.433 0.041 0.253 | 2.640 0.208  
## 5 | 4.125 | -3.515 0.043 0.726 | 0.015 0.000  
## 6 | 13.799 | 13.405 0.625 0.944 | -0.746 0.017  
## 7 | 8.193 | -0.766 0.002 0.009 | -6.309 1.189  
## 8 | 4.743 | -4.474 0.070 0.890 | 0.354 0.004  
## 9 | 6.554 | -4.995 0.087 0.581 | 1.001 0.030  
## 10 | 4.703 | -4.558 0.072 0.939 | 0.065 0.000  
## cos2 Dim.3 ctr cos2   
## 1 0.000 | 1.753 0.103 0.151 |  
## 2 0.329 | -1.210 0.049 0.023 |  
## 3 0.018 | -0.069 0.000 0.000 |  
## 4 0.150 | 2.206 0.163 0.104 |  
## 5 0.000 | -0.057 0.000 0.000 |  
## 6 0.003 | -0.714 0.017 0.003 |  
## 7 0.593 | -2.674 0.239 0.107 |  
## 8 0.006 | -0.733 0.018 0.024 |  
## 9 0.023 | -0.678 0.015 0.011 |  
## 10 0.000 | 1.002 0.034 0.045 |  
##   
## Variables  
## Dim.1 ctr cos2 Dim.2 ctr cos2   
## ClumpThickness | 2.090 10.631 0.552 | -0.106 0.235 0.001 |  
## UniformityOfCellShape | 2.683 17.509 0.816 | 0.337 2.367 0.013 |  
## MarginalAdhesion | 2.324 13.142 0.663 | -0.096 0.194 0.001 |  
## SingleEpithelialCellSize | 1.718 7.183 0.603 | 0.468 4.579 0.045 |  
## BareNuclei | 3.172 24.478 0.760 | -1.595 53.088 0.192 |  
## BlandChromatin | 2.035 10.080 0.698 | 0.083 0.144 0.001 |  
## NormalNucleoli | 2.500 15.205 0.671 | 1.287 34.592 0.178 |  
## Mitosis | 0.853 1.771 0.248 | 0.480 4.801 0.078 |  
## Dim.3 ctr cos2   
## ClumpThickness 1.769 73.114 0.395 |  
## UniformityOfCellShape 0.164 0.628 0.003 |  
## MarginalAdhesion -0.988 22.800 0.120 |  
## SingleEpithelialCellSize -0.078 0.141 0.001 |  
## BareNuclei -0.231 1.250 0.004 |  
## BlandChromatin -0.216 1.087 0.008 |  
## NormalNucleoli -0.201 0.945 0.004 |  
## Mitosis -0.038 0.034 0.001 |

dimdesc(P2, axes=c(1,2,3))#PRINT THE FIRST 3 COMPONENTS

## $Dim.1  
## $Dim.1$quanti  
## correlation p.value  
## UniformityOfCellShape 0.9032968 2.281958e-258  
## BareNuclei 0.8715686 5.960408e-218  
## BlandChromatin 0.8353417 2.771969e-183  
## NormalNucleoli 0.8192617 1.666957e-170  
## MarginalAdhesion 0.8145448 5.379777e-167  
## SingleEpithelialCellSize 0.7765335 5.840263e-142  
## ClumpThickness 0.7429172 1.270957e-123  
## Mitosis 0.4978162 4.777170e-45  
##   
##   
## $Dim.2  
## $Dim.2$quanti  
## correlation p.value  
## NormalNucleoli 0.4218628 1.551838e-31  
## Mitosis 0.2798113 4.860360e-14  
## SingleEpithelialCellSize 0.2116575 1.604034e-08  
## UniformityOfCellShape 0.1133931 2.679666e-03  
## BareNuclei -0.4382004 3.673795e-34  
##   
##   
## $Dim.3  
## $Dim.3$quanti  
## correlation p.value  
## ClumpThickness 0.62859581 3.885074e-78  
## BlandChromatin -0.08851363 1.925358e-02  
## MarginalAdhesion -0.34615202 4.163002e-21

### Conclusion and discussion

### The data is well suited for PCA;the independents variables could be reduced to essentially 3 dimensions;We run the PCA on the original data ,the scale data and also the filtered (where the highly correlated variable is removed ) => we can conclude that if our purpose is to PCA the data ,the original version of the data is better has the larger portion of the variance in the data is more explained by the first 2 components .

### From the output above in regard to the correlation between the new dimensions and the original varaible we can describe those(3 first) dimensions as :

### Prin1(Which explain 69% of the variance )

### Dimension1:(UniformityOfcelSize+uniformityofcellshape+BareNuclei+BlandChromatin+NormalNuclein+MarginalAdhesion+SingleEpithelislCellsize+ClumpThicness)...Here I retain correlation greater than 0.5

### Prin2(Wchich explain 7% of the variance)

### Dimension2:the opposite of BareNuclei ;Since the correlation are pretty low ,I retain -0.47 even its sligthly lower than 0.5

### Prin3

### Dimension3:ClumThickness

### Here we attempt to use stepwise discriminant analysis to classify observation

### Stepwise discriminant analysis

library(klaR)

## Loading required package: MASS

Step.lda<- greedy.wilks(Classes ~ ., data = filteredBcData, niveau = 0.1)  
Step.lda

## Formula containing included variables:   
##   
## Classes ~ UniformityOfCellShape + BareNuclei + ClumpThickness +   
## BlandChromatin + NormalNucleoli + SingleEpithelialCellSize +   
## MarginalAdhesion  
## <environment: 0x7fbe88487188>  
##   
##   
## Values calculated in each step of the selection procedure:   
##   
## vars Wilks.lambda F.statistics.overall  
## 1 UniformityOfCellShape 0.3293475 1419.3055  
## 2 BareNuclei 0.2207498 1228.4451  
## 3 ClumpThickness 0.1937618 963.9594  
## 4 BlandChromatin 0.1809163 785.5070  
## 5 NormalNucleoli 0.1734555 660.4521  
## 6 SingleEpithelialCellSize 0.1713396 557.7940  
## 7 MarginalAdhesion 0.1700575 481.7618  
## p.value.overall F.statistics.diff p.value.diff  
## 1 2.945621e-170 1419.305530 2.945621e-170  
## 2 4.757045e-229 342.396597 0.000000e+00  
## 3 4.003836e-247 96.802766 0.000000e+00  
## 4 6.308836e-256 49.275714 5.298650e-12  
## 5 8.774908e-261 29.807784 6.648753e-08  
## 6 3.411074e-261 8.545884 3.575947e-03  
## 7 6.296508e-261 5.209579 2.276536e-02

# Logistic regression in base R

# Here is the logistic regression approach ,we start with a full model approach (all Variable include and later we use forward and backward method)

# LOGISTIC REGRESSION :fullmodel and stepwise selection

fullmodel<- glm(Classes ~., data = filteredBcData, family = "binomial")  
summary(fullmodel)

##   
## Call:  
## glm(formula = Classes ~ ., family = "binomial", data = filteredBcData)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.5529 -0.1373 -0.0745 0.0340 2.4928   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -9.44800 0.99296 -9.515 < 2e-16 \*\*\*  
## ClumpThickness 0.52067 0.12585 4.137 3.52e-05 \*\*\*  
## UniformityOfCellShape 0.36934 0.16064 2.299 0.021495 \*   
## MarginalAdhesion 0.24181 0.10880 2.223 0.026249 \*   
## SingleEpithelialCellSize 0.07451 0.14356 0.519 0.603758   
## BareNuclei 0.32045 0.08296 3.863 0.000112 \*\*\*  
## BlandChromatin 0.42783 0.15299 2.796 0.005167 \*\*   
## NormalNucleoli 0.13119 0.09929 1.321 0.186390   
## Mitosis 0.54570 0.29725 1.836 0.066379 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 900.53 on 698 degrees of freedom  
## Residual deviance: 124.57 on 690 degrees of freedom  
## AIC: 142.57  
##   
## Number of Fisher Scoring iterations: 8

#Entering varaiable ,Intercept first  
nothing <- glm(Classes ~ 1,family=binomial,data=filteredBcData)  
summary(nothing)

##   
## Call:  
## glm(formula = Classes ~ 1, family = binomial, data = filteredBcData)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.9195 -0.9195 -0.9195 1.4593 1.4593   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.64207 0.07958 -8.068 7.12e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 900.53 on 698 degrees of freedom  
## Residual deviance: 900.53 on 698 degrees of freedom  
## AIC: 902.53  
##   
## Number of Fisher Scoring iterations: 4

#backwards = step(fullmodel) # Backwards selection is the default  
backwards = step(fullmodel,trace=0)   
#would suppress step by step output.  
formula(backwards)

## Classes ~ ClumpThickness + UniformityOfCellShape + MarginalAdhesion +   
## BareNuclei + BlandChromatin + NormalNucleoli + Mitosis

summary(backwards)

##   
## Call:  
## glm(formula = Classes ~ ClumpThickness + UniformityOfCellShape +   
## MarginalAdhesion + BareNuclei + BlandChromatin + NormalNucleoli +   
## Mitosis, family = "binomial", data = filteredBcData)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.5650 -0.1372 -0.0748 0.0342 2.4478   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -9.36410 0.97160 -9.638 < 2e-16 \*\*\*  
## ClumpThickness 0.52250 0.12593 4.149 3.34e-05 \*\*\*  
## UniformityOfCellShape 0.38971 0.15776 2.470 0.01350 \*   
## MarginalAdhesion 0.25171 0.10651 2.363 0.01811 \*   
## BareNuclei 0.32481 0.08265 3.930 8.49e-05 \*\*\*  
## BlandChromatin 0.43636 0.15229 2.865 0.00416 \*\*   
## NormalNucleoli 0.14109 0.09805 1.439 0.15015   
## Mitosis 0.54681 0.29387 1.861 0.06278 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 900.53 on 698 degrees of freedom  
## Residual deviance: 124.84 on 691 degrees of freedom  
## AIC: 140.84  
##   
## Number of Fisher Scoring iterations: 8

#back2 = glm(Classes ~ ,family=binomial,)  
#summary(back2)  
#back2$deviance  
forwards = step(nothing,scope=list(lower=formula(nothing),upper=formula(fullmodel)),direction="forward")

## Start: AIC=902.53  
## Classes ~ 1  
##   
## Df Deviance AIC  
## + UniformityOfCellShape 1 284.25 288.25  
## + BareNuclei 1 362.50 366.50  
## + BlandChromatin 1 401.35 405.35  
## + ClumpThickness 1 464.05 468.05  
## + SingleEpithelialCellSize 1 481.71 485.71  
## + NormalNucleoli 1 488.51 492.51  
## + MarginalAdhesion 1 492.55 496.55  
## + Mitosis 1 731.08 735.08  
## <none> 900.53 902.53  
##   
## Step: AIC=288.25  
## Classes ~ UniformityOfCellShape  
##   
## Df Deviance AIC  
## + BareNuclei 1 193.58 199.58  
## + BlandChromatin 1 210.68 216.68  
## + ClumpThickness 1 211.39 217.39  
## + MarginalAdhesion 1 237.99 243.99  
## + NormalNucleoli 1 247.73 253.73  
## + Mitosis 1 248.32 254.32  
## + SingleEpithelialCellSize 1 257.38 263.38  
## <none> 284.25 288.25  
##   
## Step: AIC=199.58  
## Classes ~ UniformityOfCellShape + BareNuclei  
##   
## Df Deviance AIC  
## + ClumpThickness 1 157.85 165.85  
## + BlandChromatin 1 170.21 178.21  
## + Mitosis 1 172.08 180.08  
## + NormalNucleoli 1 178.95 186.95  
## + SingleEpithelialCellSize 1 182.32 190.32  
## + MarginalAdhesion 1 184.32 192.32  
## <none> 193.58 199.58  
##   
## Step: AIC=165.85  
## Classes ~ UniformityOfCellShape + BareNuclei + ClumpThickness  
##   
## Df Deviance AIC  
## + BlandChromatin 1 139.88 149.88  
## + MarginalAdhesion 1 146.15 156.15  
## + Mitosis 1 147.14 157.14  
## + NormalNucleoli 1 148.58 158.58  
## + SingleEpithelialCellSize 1 150.49 160.49  
## <none> 157.85 165.85  
##   
## Step: AIC=149.88  
## Classes ~ UniformityOfCellShape + BareNuclei + ClumpThickness +   
## BlandChromatin  
##   
## Df Deviance AIC  
## + MarginalAdhesion 1 132.66 144.66  
## + Mitosis 1 133.40 145.40  
## + NormalNucleoli 1 136.28 148.28  
## + SingleEpithelialCellSize 1 136.75 148.75  
## <none> 139.88 149.88  
##   
## Step: AIC=144.66  
## Classes ~ UniformityOfCellShape + BareNuclei + ClumpThickness +   
## BlandChromatin + MarginalAdhesion  
##   
## Df Deviance AIC  
## + Mitosis 1 126.97 140.97  
## + NormalNucleoli 1 129.70 143.70  
## <none> 132.66 144.66  
## + SingleEpithelialCellSize 1 131.31 145.31  
##   
## Step: AIC=140.97  
## Classes ~ UniformityOfCellShape + BareNuclei + ClumpThickness +   
## BlandChromatin + MarginalAdhesion + Mitosis  
##   
## Df Deviance AIC  
## + NormalNucleoli 1 124.84 140.84  
## <none> 126.97 140.97  
## + SingleEpithelialCellSize 1 126.36 142.36  
##   
## Step: AIC=140.84  
## Classes ~ UniformityOfCellShape + BareNuclei + ClumpThickness +   
## BlandChromatin + MarginalAdhesion + Mitosis + NormalNucleoli  
##   
## Df Deviance AIC  
## <none> 124.84 140.84  
## + SingleEpithelialCellSize 1 124.57 142.57

forwards

##   
## Call: glm(formula = Classes ~ UniformityOfCellShape + BareNuclei +   
## ClumpThickness + BlandChromatin + MarginalAdhesion + Mitosis +   
## NormalNucleoli, family = binomial, data = filteredBcData)  
##   
## Coefficients:  
## (Intercept) UniformityOfCellShape BareNuclei   
## -9.3641 0.3897 0.3248   
## ClumpThickness BlandChromatin MarginalAdhesion   
## 0.5225 0.4364 0.2517   
## Mitosis NormalNucleoli   
## 0.5468 0.1411   
##   
## Degrees of Freedom: 698 Total (i.e. Null); 691 Residual  
## Null Deviance: 900.5   
## Residual Deviance: 124.8 AIC: 140.8

head(filteredBcData)

## Classes ClumpThickness UniformityOfCellShape MarginalAdhesion  
## 1 Benign 5 1 1  
## 2 Benign 5 4 5  
## 3 Benign 3 1 1  
## 4 Benign 6 8 1  
## 5 Benign 4 1 3  
## 6 Malignant 8 10 8  
## SingleEpithelialCellSize BareNuclei BlandChromatin NormalNucleoli  
## 1 2 1 3 1  
## 2 7 10 3 2  
## 3 2 2 3 1  
## 4 3 4 3 7  
## 5 2 1 3 1  
## 6 7 10 9 7  
## Mitosis  
## 1 1  
## 2 1  
## 3 1  
## 4 1  
## 5 1  
## 6 1

# OUSTIDE THE SCOPE OF OUR COURSE ,WE ATTEMPT TO USE THE MACHINES LEARNING TECHNIQUES BELOW

# MACHINE LEARNING PART MACHINE LEARNING PART

# tidyr package is an data manipulation package for easy selection similar to SQL queries

# doParallel is a package for clustering purpose

library(tidyr)

##   
## Attaching package: 'tidyr'

## The following object is masked from 'package:mice':  
##   
## complete

#BreastCancer  
# configure multicore  
library(doParallel)

## Loading required package: foreach

## Loading required package: iterators

## Loading required package: parallel

cl <- makeCluster(detectCores())  
registerDoParallel(cl)

### IT IS NOT RECOMMENDED TO USE The entire dataset for building models

### Usually you splitn the dataset into training and testing and you build the model on training and you validate your results using testing dataset.So below we attempt to take this roadWe partition our dataset into 70% training and 30% testing

library(caret)  
set.seed(100)  
BreastCancer=filteredBcData  
index <- createDataPartition(BreastCancer$Classes, p = 0.7, list = FALSE)  
training <- BreastCancer[index, ]  
testing <- BreastCancer[-index, ]

### below we atempt to visually look into the distribution of each variable and separatethe case of training and testing ;we got the plots below ;dplyr is also a data manipulation tool for group work

library(dplyr)

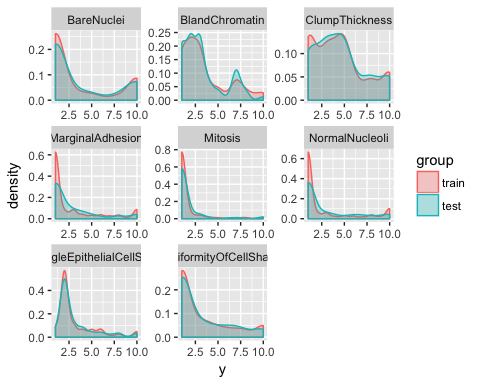
##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':  
##   
## select

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

rbind(data.frame(group = "train", training),  
 data.frame(group = "test", testing)) %>%  
 gather(x, y, ClumpThickness:Mitosis) %>%  
 ggplot(aes(x = y, color = group, fill = group)) +  
 geom\_density(alpha = 0.3) +  
 facet\_wrap( ~ x, scales = "free", ncol = 3)



set.seed(100)

### Here we attempt to use ClumThickness as a reponse varaible

#### Clump thickness: Benign cells tend to be grouped in monolayers(lower values), while cancerous cells are often grouped in multilayers(higher values) .

### We use logistic regression frm the caret package here

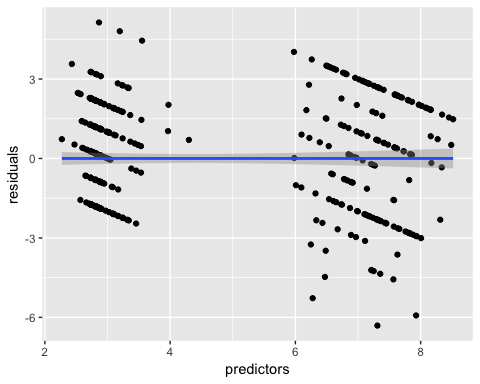
glmFit <- caret::train(ClumpThickness ~ .,  
 data = training,  
 method = "glm",  
 preProcess = c("scale", "center"),  
 trControl = trainControl(method = "repeatedcv",  
 number = 10,  
 repeats = 10,  
 savePredictions = TRUE,  
 verboseIter = FALSE))  
glmFit

## Generalized Linear Model   
##   
## 490 samples  
## 8 predictor  
##   
## Pre-processing: scaled (8), centered (8)   
## Resampling: Cross-Validated (10 fold, repeated 10 times)   
## Summary of sample sizes: 441, 442, 441, 441, 441, 441, ...   
## Resampling results:  
##   
## RMSE Rsquared   
## 1.949853 0.5378084  
##   
##

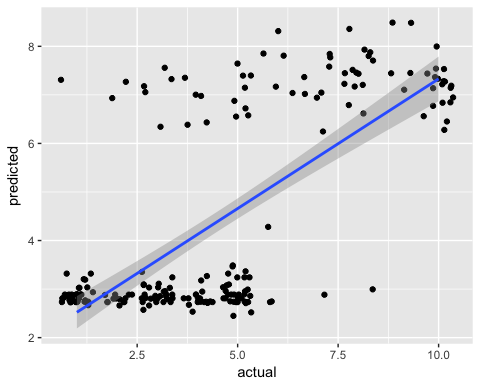
predictions <- predict(glmFit, testing)  
  
table(glmFit$finalModel$linear.predictors == glmFit$finalModel$fitted.values)

##   
## TRUE   
## 490

data.frame(residuals = resid(glmFit),  
 predictors = glmFit$finalModel$linear.predictors) %>%  
 ggplot(aes(x = predictors, y = residuals)) +  
 geom\_jitter() +  
 geom\_smooth(method = "lm")



data.frame(actual = testing$ClumpThickness,  
 predicted = predictions) %>%  
 ggplot(aes(x = actual, y = predicted)) +  
 geom\_jitter() +  
 geom\_smooth(method = "lm")



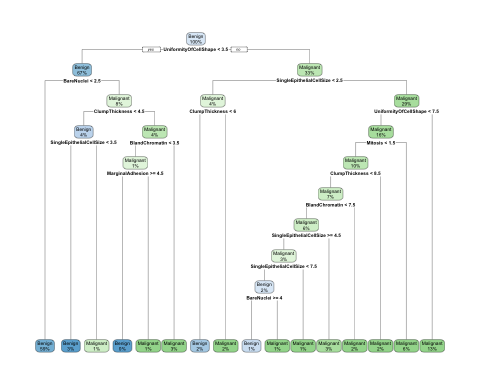
### BACK TO OUR CLASSES WE USE CLASSIFICATION HERE

### 1-DECISION TREES

### We use rpart which is a machine learning package in R

### Below one can see how the classification is done ,for each data point the algorithm used to classifify the data point follow the graph below

library(rpart)  
library(rpart.plot)  
  
set.seed(100)  
DecisionFit <- rpart(Classes ~ .,  
 data = training,  
 method = "class",  
 control = rpart.control(xval = 10,  
 minbucket = 2,  
 cp = 0),  
 parms = list(split = "information"))  
  
rpart.plot(DecisionFit, extra = 100)



### HERE WE ATTEMPT TO USE RANDOM FOREST ALGORITHM BY USING THE CARET PACKAGE AGAIN

### Since value could be ramdom ,we set the seed to 100 to be able to replicate the exact same values;We get also the confusion matrix using this technique

### 2.RANDOM FOREST

set.seed(100)  
RandomFit <- caret::train(Classes ~ .,  
 data = training,  
 method = "rf",  
 preProcess = c("scale", "center"),  
 trControl = trainControl(method = "repeatedcv",  
 number = 10,  
 repeats = 10,  
 savePredictions = TRUE,  
 verboseIter = FALSE))

## Loading required package: randomForest

## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':  
##   
## combine

## The following object is masked from 'package:ggplot2':  
##   
## margin

RandomFit$finalModel$confusion

## Benign Malignant class.error  
## Benign 313 8 0.02492212  
## Malignant 4 165 0.02366864

### We found this approach to be quite revealing ,so we decide to explore the importance of the feature using randam forest The visual results are below

### FEATURE IMPORTANCE

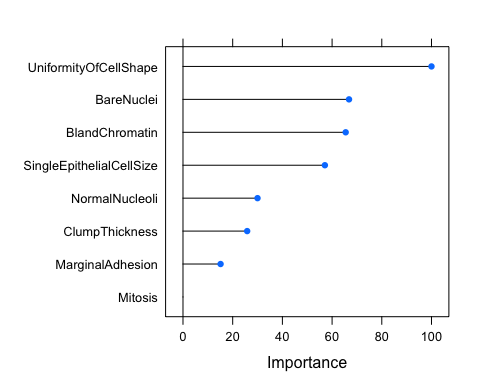
### We extensively use ggplot which is arguably the most important graphic tool in R

### We also plot the actual values (benign or Malignant ) versus the predicted .where we can see visually the misclassification results beside the corrected classification

imp <- RandomFit$finalModel$importance  
imp[order(imp, decreasing = TRUE), ]

## UniformityOfCellShape BareNuclei BlandChromatin   
## 56.250053 38.897607 38.194363   
## SingleEpithelialCellSize NormalNucleoli ClumpThickness   
## 33.816502 19.640014 17.462663   
## MarginalAdhesion Mitosis   
## 11.866657 3.955989

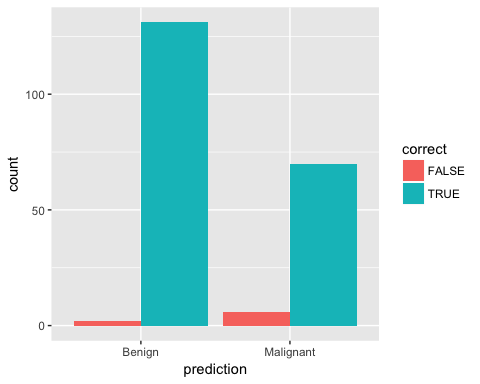
# ESTAMATION OF VARIABLE IMPORTANCE  
importance <- varImp(RandomFit, scale = TRUE)  
plot(importance)



confusionMatrix(predict(RandomFit, testing), testing$Classes)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Benign Malignant  
## Benign 131 2  
## Malignant 6 70  
##   
## Accuracy : 0.9617   
## 95% CI : (0.926, 0.9833)  
## No Information Rate : 0.6555   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.9163   
## Mcnemar's Test P-Value : 0.2888   
##   
## Sensitivity : 0.9562   
## Specificity : 0.9722   
## Pos Pred Value : 0.9850   
## Neg Pred Value : 0.9211   
## Prevalence : 0.6555   
## Detection Rate : 0.6268   
## Detection Prevalence : 0.6364   
## Balanced Accuracy : 0.9642   
##   
## 'Positive' Class : Benign   
##

results <- data.frame(actual = testing$Classes,  
 predict(RandomFit, testing, type = "prob"))  
  
results$prediction <- ifelse(results$Benign > 0.5, "Benign",  
 ifelse(results$Malignant > 0.5, "Malignant", NA))  
  
results$correct <- ifelse(results$actual == results$prediction, TRUE, FALSE)  
  
ggplot(results, aes(x = prediction, fill = correct)) +  
 geom\_bar(position = "dodge")



ggplot(results, aes(x = prediction, y = Benign, color = correct, shape = correct)) +  
 geom\_jitter(size = 3, alpha = 0.6)

